

Atividades Laboratoriais

SECOMPP & SMAT 2011

16 de Outubro de 2011

1 Algoritmo de Canny

Em 1986, John Canny definiu um conjunto de objetivos (três deles) que deveriam ser atingidos por um detector de arestas, descrevendo então um método ótimo:

Taxa de Erro: O detector de arestas deve responder apenas as arestas e encontrar todas elas sem que nenhuma seja perdida;

Localização: A distância entre os *pixels* de uma aresta e o centro da mesma aresta encontrada por um detector deve ser tão pequena quanto possível;

Quantidade de Respostas : O detector não identificaria múltiplos *pixels* na aresta onde apenas uma simples aresta existe. Em outras palavras, o número de máximos locais em torno da aresta verdadeira deve ser mínimo.

O algoritmo de Canny consiste em expressar matematicamente os três critérios citados anteriormente, bem como encontrar soluções ótimas para eles. Como é difícil encontrar algo que os satisfaçam completamente, o uso de otimizações com arestas de degrau unidimensionais que foram corrompidas por ruído branco gaussiano, levam a conclusão de que, uma aproximação interessante, para este detector ótimo, é a primeira derivada de uma gaussiana, como mostra a Equação 1.

$$G'(x) = \frac{d}{dx} e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

Supondo que $f(x, y)$ seja a imagem a ser processada e, temos então a função $G(x, y)$, ilustrada na Equação 2.

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Agora fazendo a convolução da imagem f com a função gaussiana G temos:

$$f_s(x, y) = G(x, y) \times f(x, y) \quad (3)$$

Logo após calculamos a magnitude e a direção (ângulo) do vetor gradiente:

$$M(x, y) = \sqrt{g_x^2 + g_y^2} \quad (4)$$

$$\alpha(x, y) = \text{tg}^{-1} \left[\frac{g_y}{g_x} \right] \quad (5)$$

onde:

$$g_x = \frac{\partial f_s}{\partial x} \quad \text{e} \quad g_y = \frac{\partial f_s}{\partial y} \quad (6)$$

Assumindo que d_1 , d_2 , d_3 e d_4 indicam as quatro direções (horizontal, -45° , vertical e $+45^\circ$) possíveis para uma aresta 3×3 . O esquema de supressão para esse caso consiste em:

- Encontrar a direção d_k que está mais próxima de $\alpha(x, y)$;
- Se $M(x, y)$ é menor a pelo menos um dos *pixels* de sua vizinhança (d_k), então $g_n(x, y) = 0$ ocasionando uma supressão, caso contrário $g_n(x, y) = M(x, y)$.

O $g_n(x, y)$ citado anteriormente é a imagem com a supressão de não máximos ou seja, $g_n(x, y)$ é a imagem com as bordas afinadas.

A operação de limiarização de Canny pode ser visualizada utilizando a criação de duas outras imagens:

$$g_{nH}(x, y) = g_n(x, y) \leq T_H \quad (7)$$

e

$$g_{nL}(x, y) = g_n(x, y) \leq T_L \quad (8)$$

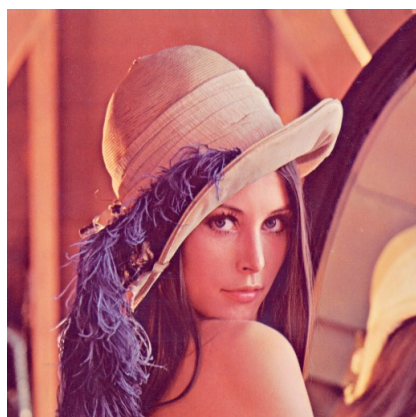
onde no começo tanto g_{nH} quanto g_{nL} irá ter apenas valores 0 e, após a limiarização, g_{nH} terá uma quantidade de *pixels* zero menor do que g_{nL} , porém, todos os *pixels* que forem diferentes de zero em g_{nH} serão adicionados a imagem g_{nL} porque a última imagem será formada com um limiar mais baixo ainda:

$$g_{nL}(x, y) = g_{nL}(x, y) - g_{nH}(x, y) \quad (9)$$

Depois de feitas as operações de limiarização, os *pixels* poderão ser classificados como fortes e fracos. Todos os *pixels* fortes de $g_{nH}(x, y)$ são marcados como *pixels* verdadeiros da aresta.

$$M(x, y) = \sqrt{I'_x(x, y)^2 + I'_y(x, y)^2} \quad (10)$$

O resultado da aplicação do algoritmo de Canny que discutimos ao longo desta seção, fizemos o teste na Figura 1(a) e, como imagem resultante obtivemos a Figura 1(b). Esta imagem também foi processada pelo sistema desenvolvido. Para os testes, utilizamos os parâmetros: $\sigma = 1$, limiar inferior = 0 e limiar superior = -1.



(a) Imagem original.



(b) $\sigma = 1$; limiar inferior = 0; limiar superior = -1

Figura 1: Resultado obtido com o algoritmo de Canny.

2 Transformadas de Fourier

A Transformada de Fourier, baptizada em homenagem a Jean-Baptiste Joseph Fourier, é uma transformada integral que expressa uma função em termos de funções de base sinusoidal, i.e., como soma ou integral de funções sinusoidais multiplicadas por coeficientes (“amplitudes”). Existem diversas variações directamente relacionadas desta transformada, dependendo do tipo de função a transformar.

2.1 Transformada contínua de Fourier

Geralmente, a denominação “Transformada de Fourier” refere-se à Transformada de Fourier para funções contínuas, que representa qualquer função integrável $f(t)$ como a soma de exponenciais complexas com frequência angular ω e amplitude complexa $F(\omega)$:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

$$f(t) = F^{-1}(F(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega$$

2.2 Transformada discreta de Fourier

Para uso em computadores, seja para aplicações científicas ou em processamento digital de sinais, é preciso ter valores x_k discretos. Para isso existe a versão da transformada para funções discretas.

$$x_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{\frac{2\pi i}{n} jk} \quad k = 0, \dots, n-1.$$

$$f_j = \sum_{k=0}^{n-1} x_k e^{\frac{2\pi i}{n} jk} \quad j = 0, \dots, n-1.$$

2.3 Algumas transformadas de Fourier

Nesta tabela, $\delta(t)$ é a delta de Dirac, e $u(t)$ é a função de passo Heaviside, sgn é a função sinal, $rect$ é a função retangular, $sinc$ é a função $sinc(x) = \frac{sen(\pi x)}{\pi x}$.

$\mathbf{f(t)}$	$\mathbf{F(\omega)}$
$\delta(t)$	1
$\delta(t-a)$	$e^{-ia\omega}$
$u(t)$	$\pi\delta(\omega) + \frac{1}{i\omega}$
1	$2\pi\delta(\omega)$
$sgn(t)$	$\frac{2}{i\omega}$
$rect(t/a)\cos(\omega_0 t)$	$a \left(sinc\left(\frac{(\omega-\omega_0)a}{2\pi}\right) + sinc\left(\frac{(\omega+\omega_0)a}{2\pi}\right) \right)$

Tabela 1: Fonte: http://pt.wikipedia.org/wiki/Transformada_de_Fourier